

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.**

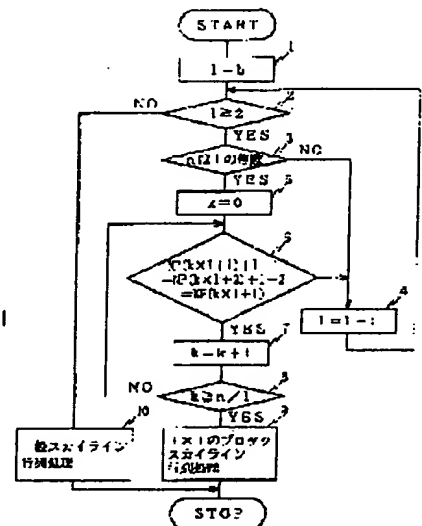
(97)

(43)Date of publication of application : 14.06.1994

**G06F 15/324**

(72)Inventor : USHIRO YASUNORI

**CONSTITUTION:** This calculating device for simultaneous linear equations calculates the solution of the simultaneous linear equations with the skyline matrix provided by the structural analysis using the finite element method provided by the structural stores the coefficient matrix A given by one-dimensional arrangement the leading address table NP of the respective columns and the right side vector B of A, and performs processing separately for the case where the heads of skylines are equal for (1) pieces (1=2, 3, ..., 6) and the other case corresponding to the value of the NP. Therefore, the parallel vector computer enables high-speed calculation 7-20 times as fast compared to the conventional system.



[Date of extinction of right]

01/10/25 19:20

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平 6 - 1 6 8 2 6 2

(43) 公開日 平成 6 年 (1994) 6 月 14 日

(51) Int. Cl. <sup>5</sup>  
G06F 15/324

識別記号 庁内整理番号  
7343-5L

F I

技術表示箇所

審査請求 未請求 請求項の数 1 (全 5 頁)

(21) 出願番号 特願平 4 - 3 1 9 7 5 6

(22) 出願日 平成 4 年 (1992) 11 月 30 日

(71) 出願人 0 0 0 0 0 5 1 0 8

株式会社日立製作所

東京都千代田区神田駿河台四丁目 6 番地

(72) 発明者 後 保範

神奈川県横浜市戸塚区戸塚町 5 0 3 0 番地

株式会社日立製作所ソフトウェア開発本

部内

(74) 代理人 弁理士 小川 勝男

(54) 【発明の名称】 連立一次方程式に関する計算装置

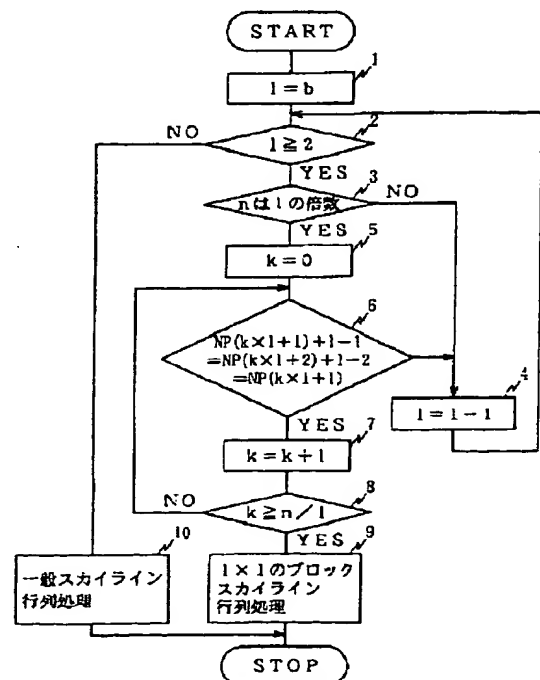
(57) 【要約】

【目的】 本発明は、並列ベクトル計算機上で、大規模な連立一次方程式の解析をするのに有効な、特に有限要素法を使用した構造解析で得られるスカイライン行列の三角分解に有効な装置を提供することにある。

【構成】 有限要素法を使用した構造解析で得られるスカイライン行列を係数とする連立一次方程式の解を計算する装置であり、一次元配列で与えられる係数行列 A、A の各列の先頭アドレステーブル NP 及び右辺ベクトル B を記憶し、NP の値により、スカイラインの先端が l 個 (l = 2, 3, ..., 6) ずつ揃ったケースとそれ以外のケースに分けて処理する連立一次方程式に関する計算装置。

【効果】 並列ベクトル計算機で、従来方式に比較し 7 ~ 20 倍の高速計算を可能にする。

図 1



## 【特許請求の範囲】

【請求項 1】有限要素法を使用した構造解析で得られるスカイライン行列を係数とする連立一次方程式の解を計算する装置であり、一次元配列で与えられる係数行列 A、A の各列の先頭アドレステーブル NP 及び右辺ベクトル B を記憶し、NP の値により、スカイラインの先頭が 1 個 ( $1 = 2, 3, \dots, 6$ ) ずつ揃ったケースとそれ以外のケースに分けて処理することを特徴とする連立一次方程式に関する計算装置。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】本発明は、連立一次方程式の数値解を計算する装置に係わり、特に有限要素法を使用した構造解析で得られるスカイライン行列を係数とする行列の三角分解をする装置であり、大規模な数値シミュレーションを行うベクトル計算機、並列処理方式の計算機などに関する。

## 【0002】

【従来の技術】従来の技術としては、小国 力編著「行列計算ソフトウェア、-WS、スーパーコン、並列計算機」(丸善)論じられているスカイライン行列の改訂コレスキー分解を用いた方法がある。この方式は一行一列の内積計算処理を基本としている。

## 【0003】

【発明が解決しようとする課題】前記、一行一列の内積計算を使用した改訂コレスキー分割では、並列ベクトル計算機に適用する場合に下記 3 つの問題点がある。

【0004】(1)内積演算はその結果を次に使用するため、ベクトル演算とスカラ演算が並列実行できない。

【0005】(2)この方式の内積演算では、一つ前に計算した結果を次の内積計算に使用するため、内積計算の並列実行は困難。

【0006】(3)一行一列の内積計算のため、乗算と加算を一回行うのに、2 回のベクトル・ロードが必要であり、乗加算とベクトル・ロードが同一回数できるベクトル計算機では、乗加算器が半分使用されない。

【0007】上記(1)及び(2)を解決する方法として、ガウス消去法を採用した。ガウス消去法では、ベクトル演算とスカラ演算の並列実行及び各列の消去計算の並列実行が可能である。上記(3)を解決する方法として、ブロック・スカイライン行列か、一般スカイライン行列かの判定処理を追加し、ブロック・スカイライン行列の場合は 1 行 1 列のガウス消去法を、一般スカイライン行列の場合は 1 行 1 列のガウス消去法を適用し、ベクトル演算回数に対して、ベクトル・ロード及びストアの回数の減少をさせる。

## 【0008】

【課題を解決するための手段】有限要素法を使用した構造解析で得られるスカイライン行列を係数とする行列の三角分解を、並列・ベクトル計算機で効率的に行う目的

のため下記のような技術的手段を採用した。

【0009】まず、スカイライン行列の形を次の二つに分類し、その形に合った処理方式を採用した。分類はスカイライン行列の先端が 1 個 ( $1 = 2, 3, \dots, 6$ ) ずつ揃ったケースとそれ以外のケースとにした。前者のケースをブロック・スカイライン行列、後者のケースを一般スカイライン行列と名付ける。一次元配列で与えられる係数行列 A の各列の先頭アドレステーブルを NP とし、次元数を n とするとき、 $1 \times 1$  のブロック・スカイライン行列とは次の条件が成立する場合で、他の場合は一般スカイライン行列と判定する。ブロック・スカイライン行列の条件は次元数 n が 1 で割り切れ、 $NP(k \times 1 + 1) + 1 - 1 = NP(k \times 1 + 2) + 1 - 2 = \dots = NP(k \times 1 + 1)$  なる等式が  $k = 0, 1, \dots, n/1 - 1$  のすべてで成立することである。各列の演算の並列実行、ベクトル演算とスカラ演算の並列実行及びベクトル・ロードとストアを演算回数に比較して減少させるために上記で分類した各ケースごとに次のような技術的手段を採用する。

【0010】 $1 \times 1$  ブロック・スカイライン行列の場合は、行列 A の各行の先頭アドレステーブル NP から 1 行単位にガウス消去する範囲テーブル KL を作成し、この KL テーブルに従って、1 行 1 列単位にガウス消去法を適用して三角分解を行う。

【0011】一般スカイライン行列の場合は、行列 A の各行の先頭アドレステーブル NP から一行単位にガウス消去する範囲テーブル KL を作成し、次に 1 行 (3, 4, 5, 6) 単位にガウス消去法を適用したとき、無駄計算となる要素数 P を算出し、 $P/1$  が最小となる行数 1 を求め、この KL テーブルに従って 1 行 1 列単位にガウス消去法を適用して三角分解を行う。

## 【0012】

【作用】スカイライン行列を自動的に二つに分類するのは、ブロック・スカイライン行列は特に、並列ベクトル計算機で効率良く三角分解できるためであり、一般スカイライン行列はその特性に合わせて、並列ベクトル計算機で効率良く三角分解できるためである。

【0013】 $1 \times 1$  のブロック・スカイライン行列において、行列 A の各行の先頭アドレステーブル NP から 1 行単位にガウス消去する範囲テーブルを作成するのは、これにより毎回ガウス消去する範囲の検索を不要とし効率を上げるためである。1 行 1 列単位にガウスの消去法を適用するのは、並列ベクトル計算機において下記 3 つの効果がある。

【0014】(1)1 列単位のガウス消去が並列演算可能となる。

【0015】(2)ベクトル演算とスカラ演算が並列に実行される。

【0016】(3)乗加算演算に比較して、ベクトル・ロード及びストアの回数を減少できる。(この効果は一般スカイラインより大きい)

一般スカイライン行列において、行列 A の各行の先頭アドレステーブル N P から一行単位にガウス消去する範囲テーブルを作成するのは、これにより毎回ガウス消去する範囲の検索を不要とし効率を上げるためである。次に 1 行 ( 3 , 4 , 5 , 6 ) 単位にガウス消去法を適用したとき無駄計算となる要素数 P を算出し、 $P/l$  が最小となる l を求めるのは、下記に示す ( 3 ) の効果 ( l が大きいほど大 ) と、無駄計算の数バランスを考慮して、効率をできるだけ良くするためである。1 行単位にガウスの消去法を適用するのは、並列ベクトル計算機において下記 3 つの効果がある。

【 0 0 1 7 】 ( 1 ) 1 列単位にガウス消去が並列演算可能となる。

【 0 0 1 8 】 ( 2 ) ベクトル演算とスカラー演算が並列に実行される。

【 0 0 1 9 】 ( 3 ) 乗加算演算に比較して、ベクトル・ロード及びストアの回数を減少できる。(この効果は l の値が大きいほど大である。)

【 0 0 2 0 】

【実施例】以下、本発明について図面を参照して説明する。

【 0 0 2 1 】 図 1 は本発明の基本と処理フローチャート、図 2 は  $l \times l$  のブロック・スカイライン行列の処理フローチャート、図 3 は一般スカイライン行列の処理フローチャート、図 4 は  $3 \times 3$  のブロック・スカイライン行列の構成図、図 5 は  $l \times l$  のブロック・スカイライン行列で範囲テーブル K L を求める処理フローチャート、図 6 は一般スカイライン行列で無駄計算となる要素数 P の算出例である。

【 0 0 2 2 】 図 1 の 1 は、構造解析の一節点の最大自由度数 6 を l にセットし、2 はブロック・スカイライン行列の検索か一般スカイライン行列かを判定するブロックであり、3 は  $l \times l$  ブロック・スカイラインの可能性があるかどうかを判定するブロックであり、可能性がない場合は 4 で l の値を 1 減少して 2 の判定処理にもどる。5 , 7 , 8 は  $k = 0 , 1 \dots , n/l - 1$  まで 6 のブロック・スカイライン行列の判定処理をくり返すループである。6 は  $l \times l$  のブロック・スカイライン行列の判定部で、上記くり返し中一回でも条件が成立しないと可能性なしとして 4 に行く。上記くり返し中で 6 の条件が毎回成立した場合は 9 の  $l \times l$  のブロック・スカイライン行列処理を実行する。2 の判定ブロックで条件が成立しないと 1 0 の一般スカイライン行列の処理を実行する。

【 0 0 2 3 】 図 2 の 1 1 は 1 行単位のガウス消去範囲テーブル K L の作成で、この詳細は図 5 で示す。このときセットされた K L の値を 1 6 に示す。1 2 は 1 行単位の処理ループを示す。1 3 は 1 行 1 列のガウス消去の準備で、1 7 で示すような消去必要ブロックは l を、不要ブロックは 0 を I W にセットする。次に 1 8 で示すワーク W に行列 A の 1 行の値をセットする。1 8 は  $l = 3$  とし

た例である。次に W の一行目で 2 , ... , l 行を消去、二行目で 3 , ... , l 行を消去と続けて W の消去をする。このとき対角に対応する W の値の逆数を対角行列 0 に 1 個セットする。次で、1 8 の W の値にマイナス符号をつけて 1 9 の T に移す。そして W の 1 行に作成した D の値を乗算する。続いて W の 1 行の値を対応する行列 A に移す。1 4 は主消去で 1 行の T と 1 行の W を乗算し、対応する行列 A の値に加えるこのとき 1 列単位の処理をする。このとき I W の値が 0 になる位置の処理はスキップする。本処理 2 0 の範囲に対して行う。( 3 行 3 列の例 ) 1 5 は残消去で 2 1 の範囲に対して、主消去と同様の処理をする。またここで、1 6 の K L は 1 2 の 2 回目の処理では、ガウス消去する範囲が 1 8 列目までであることを示している。

【 0 0 2 4 】 図 3 の 2 2 は 1 行単位のガウス消去範囲テーブル K L の作成をする。2 3 は同時処理する行数 l の決定で、図 6 に示す P の値で算出し、 $n_1$  は n より小さい最大の l の倍数とする。2 4 は 1 行単位の処理ループを示す。2 5 は 1 行 1 列のガウス消去の準備で、図 2 の 1 3 と同様な処理をする。2 6 は主消去で 1 行の T と 1 行の W を乗算し、対応する A の値に加える。このとき 1 列単位に処理をするのが図 2 の 1 4 と異なる。2 7 は最後に残った  $l - 1$  行以下の処理ループを示す。2 8 はこの部分の処理は少ないため、一般に用いられている 1 行 1 列のガウス消去を適用する。

【 0 0 2 5 】 図 4 の 2 9 は 3 行 3 列ブロック・スカイライン行列の一例である。3 0 は k 列の対角の位置が N P ( k ) であることを示す。A ( N P ( k ) ) が k 列の対角値を示す。3 1 は k 列のスカイラインの先端の位置が N P ( k + 1 ) - 1 であることを示す。3 2 は行列 A の各列の先頭アドレステーブル N P の例である。N P の配列は次元数 n より 1 大きい。

【 0 0 2 6 】 図 5 の 3 3 は  $l \times l$  のブロック・スカイライン行列の K L テーブルの作成の準備である。3 4 は次元数 n から 1 まで - 1 ずつ逆に探索するためのループである。3 5 は i 行における K L テーブルの開始位置の計算である。3 6 , 3 7 , 3 8 , 3 9 で K L テーブルをセットする。

【 0 0 2 7 】 図 6 の 4 0 は一般スカイライン行列の 1 列である。4 1 は 3 行単位に処理する場合に発生する無駄計算となる要素を示したものである。4 2 は本例 (  $l = 3$  ) で無駄計算の要素数 P が 1 5 となり、 $P/l$  が 5 となることを示す。4 3 は 4 行単位に処理する場合に発生する無駄計算となる要素を示したものである。4 4 はこのとき無駄計算の要素数 P は 2 1 となり、 $P/l$  が 5.25 となることを示す。一般スカイライン行列において、1 行単位 (  $l = 3 , 4 , 5 , 6$  ) とする l の値の決定は  $P/l$  の値が最も小さい l を選定する。

【 0 0 2 8 】

【発明の効果】 H I T A C S - 3800 / 180 のようなベ

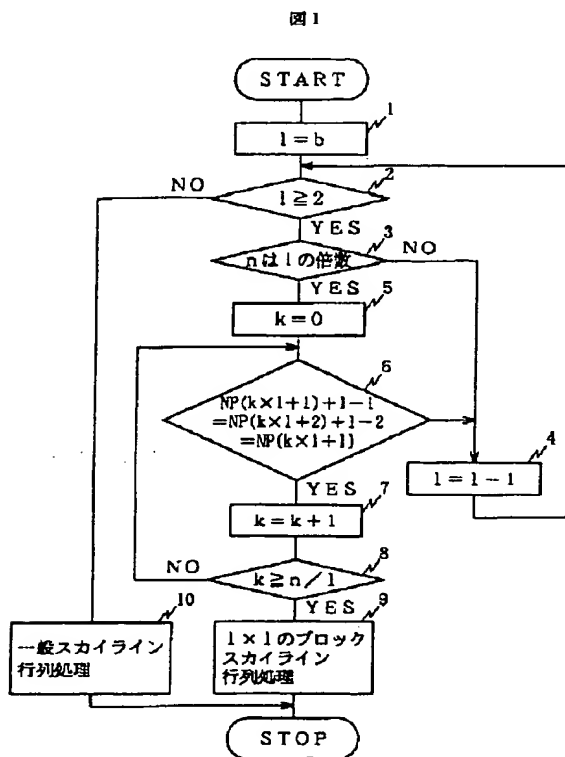
クトル計算機では、従来の計算方式に比較して、ブロック・スカイライン行列の場合の本発明の効果はCPU時間が $1/4 \sim 1/7$ に短縮し、一般スカイライン行列の場合は $1/3 \sim 1/5$ に短縮するという性能上の効果がある。

【0029】HITAC S-3800/480のように並列ベクトル計算機では、従来の計算方式に比較して、ブロック・スカイライン行列の場合の本発明の効果はCPU時間が $1/10 \sim 1/20$ に短縮し、一般スカイライン行列の場合は $1/7 \sim 1/15$ に短縮するという性能上の効果がある。

【図面の簡単な説明】

【図1】処理フローチャートで、ブロック・スカイライ

【図1】



ン行列か、一般スカイライン行列の判定方式を示す図である。

【図2】処理フローチャートで、ブロック・スカイライン行列の三角分解方式を示す図である。

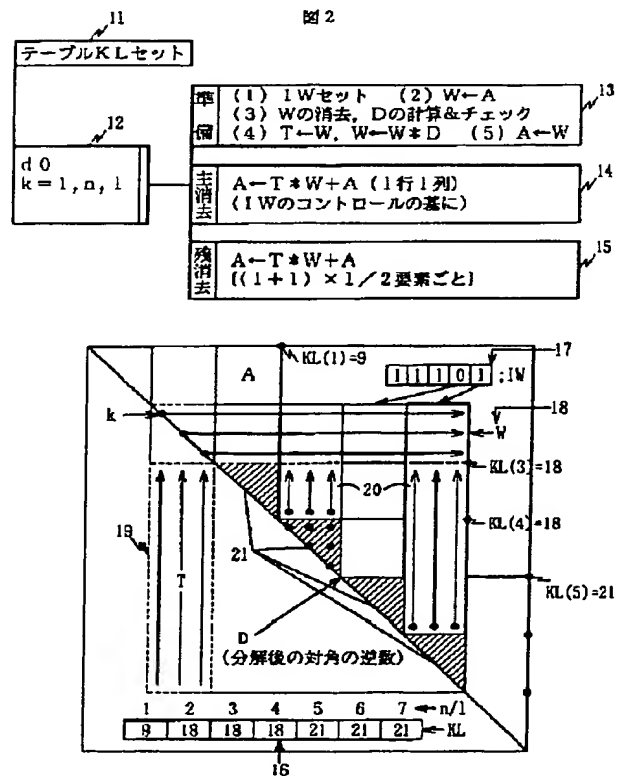
【図3】処理フローチャートで、一般スカイライン行列の三角分解方式を示す図である。

【図4】ブロック・スカイライン行列の構成で、行列Aの先頭アドレステーブルNPの例を示す図である。

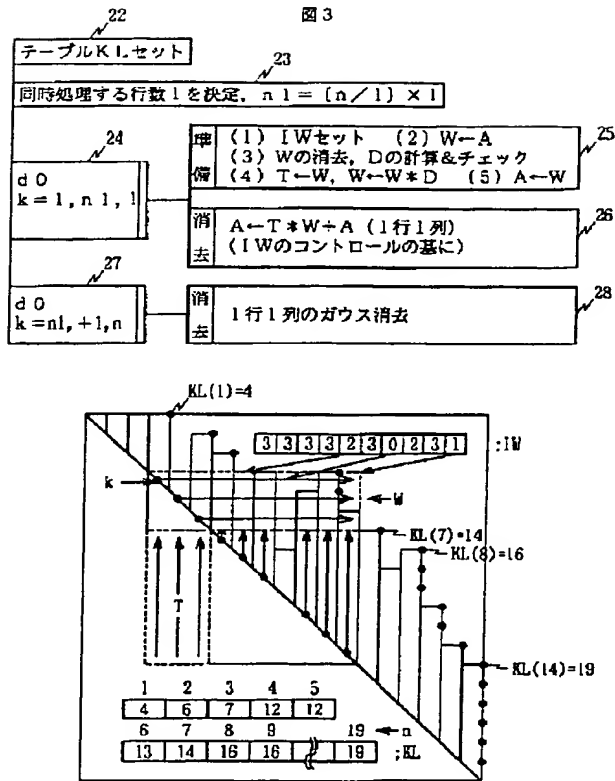
【図5】処理フローチャートで、ブロック・スカイライン行列のKLテーブルの作成方式を示す図である。

【図6】一般スカイライン行列の構成で、無駄計算となる要素数Pの算出例を示す図である。

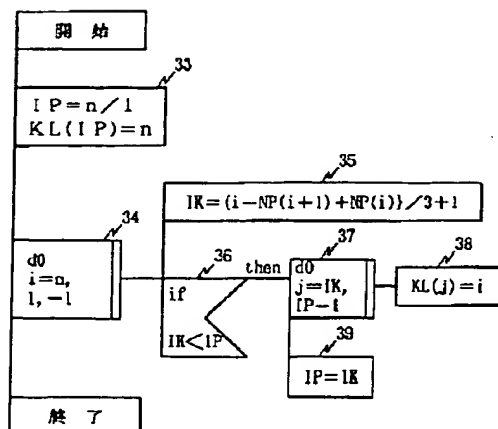
【図2】



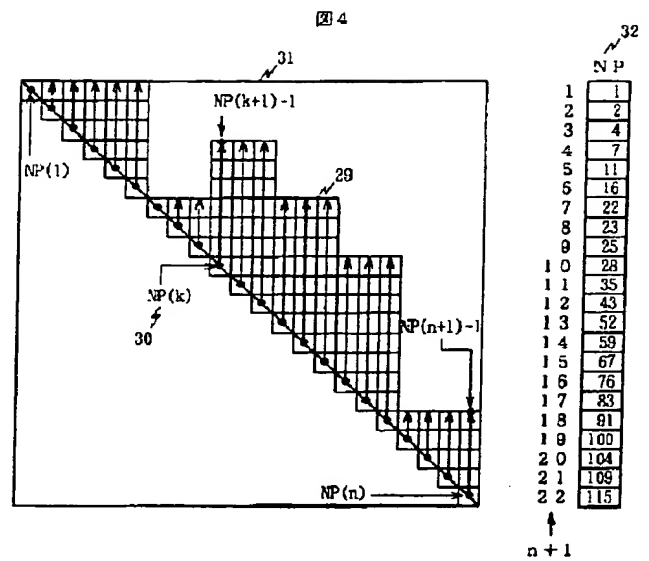
【 図 3 】



【 図 5 】



【 図 4 】



【 図 6 】

